WEBSITE DEV – Django

1. Draw the website system diagram – showing urls (routing), views & html links, and Database.

- Draw a Class digram or a function diagram detailing how the engine will operate.

2. Separate the backend and frontend – reason freely about the backend (define it and work on it completely decoupled from the front end). The backend creates and engine for the site, and it could be any piece software running on the server computer.

- It could be written in any langauge, a mixer of langauges, etc. If it is a computationally intensive piece of code, it can leverage hardware accelerators, like GPUs. In essence, it can be anything on the server machine.

3. The frontend is mostly about user experience. Likewise it has to be decoupled from the backend but since it serves the backend its link to the backend has to be clearly articulated. Its like it collects user information and experiences and routes it to the appropriate handler in the backend.

- Its appearance has to be designed independently of the backend and routing channels.
- It consits of buttons, forms, adverts most especially to instruct the user of what they want and the backend will serve the course. Backend services can have any naming conversion on the fron end.

4. Deployment: Nginx – caching, proxy, workers, etc
Gunicorn – workers, sockets, logs, etc
Supervisor – daemon to run/manage Gunicorn
Mysql database – serializable isolation level, and atomic queries
Django – pdb debug, caching, sessions, middleware, templates, etc

It is crucial to understand the flow of information – a request is received, then middlware filters it, then it gets received by the appropriate view where it can interact with the database in all sorts of ways, perform complex manipulations and then rendered on a template, and response is created which passes via middleware to the browser.

5. AWS – EC2 instance: Configure the **swap space** – it improves performance.

- Stop all unncessary daemons and processes.
- Configure the security group to all traffic: hhtp, https, ssh and mysql.
- Minitor the server – its memory consumption and CPU load for the processes mentioned above, particularly.
- Configure curl to access most frequent pages using Crontabs - to keep the cache warm and avoid cache misses.